

# Descriptor learning for omnidirectional image matching

Jonathan Masci<sup>1,2,3</sup>

jonathan@idsia.ch

Davide Migliore<sup>1,4</sup>

davide.migliore@gmail.com

Michael M. Bronstein<sup>2</sup>

michael.bronstein@usi.ch

Jürgen Schmidhuber<sup>1,2,3</sup>

juergen@idsia.ch

<sup>1</sup>Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Manno, Switzerland

<sup>2</sup>Faculty of Informatics, Università della Svizzera Italiana (USI), Lugano, Switzerland

<sup>3</sup>Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Lugano, Switzerland

<sup>4</sup> Evidence Srl, Pisa, Italy

## Abstract

*Feature matching in omnidirectional vision systems is a challenging problem, mainly because complicated optical systems make the theoretical modelling of invariance and construction of invariant feature descriptors hard or even impossible. In this paper, we propose learning invariant descriptors using a training set of similar and dissimilar descriptor pairs. We use the similarity-preserving hashing framework, in which we are trying to map the descriptor data to the Hamming space preserving the descriptor similarity on the training set. A neural network is used to solve the underlying optimization problem. Our approach outperforms not only straightforward descriptor matching, but also state-of-the-art similarity-preserving hashing methods.*

## 1. Introduction

Feature-based matching between images has become a standard approach in computer vision literature in the last decade, in many respects due to the introduction of stable and invariant feature detection and description algorithms such as SIFT [23] and similar methods [27, 2, 38]. The usual assumption guiding the design of feature descriptors is invariance across viewpoints, which should guarantee that the same feature appearing in two different views has the same descriptor. Since perspective transformations are approximately locally affine, it is common to construct affine-invariant descriptors [21].

While being a good model in many cases, affine invariance is not sufficiently accurate in cases of wide baseline (very different view points) or even more complicated setting of optical imperfections such as

lens distortions, blur, etc. In particular, in omnidirectional vision systems the distortion is introduced intentionally (e.g., using a parabolic mirror [25]) to allow a 360° view. Designing invariant descriptors for such cases is challenging, as the invariance is complicated and cannot be easily modeled.

An alternative to ‘invariance-by-construction’ approaches which rely on a simplified invariance model is to *learn* the descriptor invariance from examples. Recent work of Strecha *et al.* [35] showed very convincingly that such approaches can significantly improve the performance of existing descriptors.

In this paper, we consider the learning of invariant descriptors for omnidirectional image matching. We construct a training set of similar and dissimilar descriptor pairs including strong optical distortions, and use a neural network to learn a mapping from the descriptor space to the Hamming space preserving similarity on the training set. Experimental results show that our approach outperforms not only straightforward descriptors, but also other similarity-preserving hashing methods. The latter observation is explained by the suboptimality of existing approaches which solve a simplified optimization problem.

The main contribution of this paper is two-fold. First, we formulate a new similarity-sensitive hashing algorithm. Second, we use this approach to learn smaller invariant descriptors suitable for feature matching in omnidirectional images. The rest of the paper is organized as follows. In Section 2, we overview the related works. Section 3 is dedicated to metric learning and similarity-preserving hashing methods. In Section 4, we describe our NNhash approach. Section 5 contains experimental results. Finally, Section 6 discusses potential future work and concludes the paper.

## 2. Background

Although feature-based correspondence problems have been investigated in depth for standard perspective cameras, omnidirectional image matching still remains an open problem, largely because of the complicated geometry introduced by lenses and curved mirrors. Broadly speaking, the existing approaches either try to reduce the problem to the simpler perspective setting, or design special descriptors suitable for omnidirectional images.

Svoboda *et al.* [36] proposed to use adaptive windows around interest points to generate normalized patches with the assumption that the displacement of the omnidirectional system is smaller than the depth of the surrounding scene. Nayar [28] showed that, given the mirror parameters, it is possible to generate a perspective version of the omnidirectional image and Mauthner *et al.* [24] used this approach to generate perspective representation of each interest point region. This unwarping procedure removes the non-linear distortions and enables the use of algorithms designed for perspective cameras. Micusik and Pajdla [26] checked the candidate correspondences between two views using the RANSAC algorithm and the epipolar constraint [13]. Construction of scale-space by means of diffusion on manifolds was used in [3, 16, 11] for the construction of local descriptors. Puig *et al.* [29] integrated the sphere camera model with the partial differential equations on manifolds framework.

Another possible solution is to consider different kind of features to exploit particular invariance in omnidirectional systems, for example, extracting one-dimensional features [5] or vertical lines [32] and defining descriptors suitable for omnidirectional images.

More recently, it was shown in [35] that one can approach the design of invariant descriptors from the perspective of *metric learning*, constructing a distance between the descriptor vectors from a training set of similar and dissimilar pairs [1, 42]. In particular, *similarity-preserving hashing* methods [14, 34, 43, 22, 30] were found especially attractive for descriptor learning, as they significantly reduce descriptor storage and comparison complexity. These methods have also been applied to image search

[17, 39, 19, 18, 20, 41], video copy detection [7], and shape retrieval [6].

In [31], binary codes were produced using a restricted Boltzmann machine and in [43] using spectral hashing in an unsupervised setting. The authors showed that the learnt binary vectors capture the similarities of the data. With such an approach it is however impossible to explicitly provide information about data similarities. Since in our problem it is easy to produce labeled data, supervised metric learning is advantageous.

### 3. Similarity preserving hashing

Given a set of keypoint descriptors, represented as  $n$ -dimensional vectors in  $\mathbb{R}^n$ , the problem of *metric learning* is to find their representation in some metric space  $(\mathbb{Z}, d_{\mathbb{Z}})$  by means of a map of the form  $\mathbf{y} : \mathbb{R}^n \rightarrow (\mathbb{Z}, d_{\mathbb{Z}})$ . The metric  $d_{\mathbb{Z}} \circ (\mathbf{y} \times \mathbf{y})$  parametrizes the similarity between the feature descriptors, which may be difficult to compute in the original representation. Typically,  $(\mathbb{Z}, d_{\mathbb{Z}})$  is fixed and  $\mathbf{y}$  is the map we are trying to find in such a way that, given a set  $\mathcal{P}$  of pairs of descriptors from corresponding points in different images (*positives*) and a set  $\mathcal{N}$  of pairs of descriptors from different points (*negatives*), we have  $d_{\mathbb{Z}}(\mathbf{y}(\mathbf{x}), \mathbf{y}(\mathbf{x}^+)) \approx 0$  for all  $(\mathbf{x}, \mathbf{x}^+) \in \mathcal{P}$  and  $d_{\mathbb{Z}}(\mathbf{y}(\mathbf{x}), \mathbf{y}(\mathbf{x}^-)) \gg 0$  for all  $(\mathbf{x}, \mathbf{x}^-) \in \mathcal{N}$  with high probability.

A particular setting of this problem, where  $\mathbb{Z} = \{\pm 1\}^m$  is the  $m$ -dimensional space of binary strings and  $d_{\mathbb{H}^m}(\mathbf{y}, \mathbf{y}') = \frac{m}{2} - \frac{1}{2} \sum_{i=1}^m \text{sign}(y_i y'_i)$  is the Hamming metric, the problem is referred to as *similarity-preserving hashing*. Here, we limit our attention to affine embeddings of the form

$$\mathbf{y} = \text{sign}(\mathbf{P}\mathbf{x} + \mathbf{t}) , \quad (1)$$

where  $\mathbf{P}$  is an  $m \times n$  matrix and  $\mathbf{t}$  is an  $m \times 1$  vector. Our goal is to find such  $\mathbf{P}$  and  $\mathbf{t}$  that minimize one of the following cost functions,

$$\begin{aligned} L_c(\mathbf{P}, \mathbf{t}) &= \mathbb{E}\{\mathbf{y}(\mathbf{x})^T \mathbf{y}(\mathbf{x}^-) - \alpha \mathbf{y}(\mathbf{x})^T \mathbf{y}(\mathbf{x}^+)\}, \text{ or} \\ L_d(\mathbf{P}, \mathbf{t}) &= \mathbb{E}\{\alpha \|\mathbf{y}(\mathbf{x}) - \mathbf{y}(\mathbf{x}^+)\|^2 - \|\mathbf{y}(\mathbf{x}) - \mathbf{y}(\mathbf{x}^-)\|^2\} \end{aligned}$$

for  $(\mathbf{x}, \mathbf{x}^+) \in \mathcal{P}$  and  $(\mathbf{x}, \mathbf{x}^-) \in \mathcal{N}$ . Both cost functions try to map positives as close as possible to each other (expressed as large correlations or small distance), and negatives as far as possible from each other (small correlation or large distance), in order to ensure low false positive (FPR) and false negative (FNR) rates.  $\alpha > 0$  is a parameter determining the tradeoff between the FPR and FNR. In practice, the expectations are approximated as means on some sufficiently large training set.

The problem  $\min_{\mathbf{P}, \mathbf{t}} L(\mathbf{P}, \mathbf{t})$  is a non-linear non-convex optimization problem without an obvious simple solution. It is commonly approached by the following two-stage relaxation: first, approximate the map  $\mathbf{y} \approx \mathbf{P}\mathbf{x}$  by removing the sign and the offset vectors, minimizing

$$\begin{aligned} \hat{L}_c(\mathbf{P}) &= \mathbb{E}\{(\mathbf{P}\mathbf{x})^T (\mathbf{P}\mathbf{x}^-) - \alpha (\mathbf{P}\mathbf{x})^T (\mathbf{P}\mathbf{x}^+)\}, \text{ or} \\ \hat{L}_d(\mathbf{P}) &= \mathbb{E}\{\alpha \|\mathbf{P}(\mathbf{x} - \mathbf{x}^+)\|^2 - \|\mathbf{P}(\mathbf{x} - \mathbf{x}^-)\|^2\} \end{aligned}$$

w.r.t. to  $\mathbf{P}$  (introducing some regularization, e.g.,  $\mathbf{P}^T \mathbf{P} = \mathbf{I}$ , in order to avoid a trivial solution  $\mathbf{P} = 0$ ). Second, fix  $\mathbf{P}^* = \arg \min_{\mathbf{P}} \hat{L}(\mathbf{P})$  and solve  $\mathbf{t}^* = \arg \min_{\mathbf{t}} L(\mathbf{P}^*, \mathbf{t})$  w.r.t.  $\mathbf{t}$ . To further simplify the problem, it is also common to assume *separability*, thus solving independently for each dimension of the hash.

### 3.1. Similarity-sensitive hashing (SSH)

In [34], the above strategy was used for the approximate minimization of the cost  $L_c$ . The computation of optimal parameters  $\mathbf{P}$  and  $\mathbf{t}$  was posed as a boosted binary classification problem, where  $d_{\mathbb{H}}(\mathbf{y}, \mathbf{y}')$  acts as a strong binary classifier, and each dimension of the linear projection  $\text{sign}(\mathbf{p}_i \mathbf{x} + t_i)$  is considered a weak classifier (here,  $\mathbf{p}_i$  denotes the  $i$ th row of  $\mathbf{P}$ ). This way, AdaBoost can be used to find a greedy approximation of the minimizer of  $L_c$  by progressively constructing  $\mathbf{P}$  and  $\mathbf{t}$ . At the  $i$ -th iteration, the  $i$ -th row of the matrix  $\mathbf{P}$  and the  $i$ -th element of the vector  $\mathbf{t}$  are found minimizing a weighted version of  $L_c$ . Since the problem is non-linear, such an optimization is a challenging problem. In [34], random projection directions were used. A better method for projection selection similar to linear discriminative analysis (LDA) was proposed [7, 9]. Weights of false positive and false negative pairs are increased, and weights of true positive and true negative pairs are decreased, using the standard AdaBoost reweighting scheme [12].

### 3.2. Covariance difference hashing (diff-hash)

In [35], it was observed that the minimization  $\min_{\mathbf{P}} \hat{L}_c(\mathbf{P})$  can be written as

$$\min_{\mathbf{P}} \text{tr}\{\mathbf{P}(\alpha \mathbf{C}_+ - \mathbf{C}_-)\mathbf{P}^T\} \text{ s.t. } \mathbf{P}^T \mathbf{P} = \mathbf{I}, \quad (2)$$

where  $\mathbf{C}_{\pm} = \mathbb{E}\{(\mathbf{x} - \mathbf{x}^{\pm})(\mathbf{x} - \mathbf{x}^{\pm})^T\}$  are the covariance matrices of the differences of the positive and negative pairs of vectors. Requiring an orthonormal projection matrix  $\mathbf{P}$ , the problem has a closed-form solution consisting of the  $m$  smallest eigenvectors of  $(\alpha \mathbf{C}_+ - \mathbf{C}_-)$ , and is thus also a separable problem. Once the projection is found in this way, the threshold vector  $\mathbf{t}$  maximizing the sum of the false positive and false negative rates is selected. This second stage also turns out separable in each dimension. In [8], a more generic kernelized version of diff-hash (kdiff-hash) was shown.

### 3.3. LDAHash

A similar method was derived in [35] by transforming the coordinates as  $\mathbf{C}_-^{-1/2} \mathbf{x}$ , which allows to write  $\min_{\mathbf{P}} \hat{L}_d(\mathbf{P})$  as

$$\min_{\mathbf{P}} \text{tr}\{\mathbf{P}(\mathbf{C}_+ \mathbf{C}_-^{-1})\mathbf{P}^T\} \text{ s.t. } \mathbf{P}^T \mathbf{P} = \mathbf{I}. \quad (3)$$

This approach resembles linear discriminant analysis (LDA), hence the name LDAhash. Requiring an orthonormal projection matrix  $\mathbf{P}$ , the problem has a separable closed-form solution consisting of the  $m$  smallest eigenvectors of  $(\mathbf{C}_+ \mathbf{C}_-^{-1})$ .

## 4. Neural network hashing (NNhash)

The problem of existing and most successful similarity-preserving hashing approaches such as LDA- or diff-hash is that they do not solve the optimization problem  $\min_{\mathbf{P}, \mathbf{t}} L(\mathbf{P}, \mathbf{t})$  but rather its relaxation. As a result, the parameters  $\mathbf{P}^*, \mathbf{t}^*$  found by these methods in the aforementioned two-stage separable scheme is suboptimal, i.e.,  $L(\mathbf{P}^*, \mathbf{t}^*) > \min L$ . Our experience shows that in some cases, the suboptimality is dramatic (at least an order of magnitude).

A way of solving the ‘true’ optimization problem is by formulating it in the neural network (NN) framework and exploiting numerous optimization techniques and heuristics developed in this field. Since

we have a way of cheaply producing labeled data, we decide to adopt the *siamese network* architecture [33, 15] which, contrary to conventional models, receives two input patterns and minimize a loss function similar to equation (2),

$$L_{nn}(\mathbf{P}, \mathbf{t}) = \frac{1}{2} \|\mathbf{y}(\mathbf{x}) - \mathbf{y}(\mathbf{x}^+)\|^2 + \frac{1}{2} (\max\{0, m - \|\mathbf{y}(\mathbf{x}) - \mathbf{y}(\mathbf{x}^-)\|\})^2, \quad (4)$$

where the constant  $m$  represents the margin between dissimilar pairs. The margin is introduced as regularization to avoid the system from minimizing the loss just pulling two vector as far apart as possible. The embedding is then learned to make positive pairs as close as possible and negative pairs at least at distance  $m$ .

Network architecture of this type can be traced back to the work of Schmidhuber and Prelinger [33] on problems of predictable classification. In [15], siamese networks were used to learn an invariant mapping of tiny images directly from pixel representation, whereas in [37] a similar approach is used to learn a model that is highly effective at matching people in similar pose which exhibits invariance to identity, clothing, background, lighting, shift and scale. An advantage of such architecture is that one can create arbitrarily complex embeddings by simply stacking many layers in the network. In all our experiments, in order to make a fair comparison to other hashing methods, we adopt a simple single layer architecture, wherein  $\mathbf{y}(\mathbf{x}) = \text{sign}(\mathbf{P}\mathbf{x} + \mathbf{t})$ . Network training attempts to find  $\mathbf{P}, \mathbf{t}$  that minimize  $L_{nn}$  (which is a regularized version of  $L_d$ ). Since we solve a non-linear problem without introducing any simplification or relaxation, the results are expected to be better compared to hashing methods described in Section 3. In the following, we refer to our method as *NNhash*.

Since a binary output is required, we adopt  $\tanh(\beta t) \approx \text{sign}(t)$  as the non-linear activation function for our siamese network, which enforces binary vectors when either  $m$  or the steepness  $\beta$  of the function is increased. Since the problem is highly non-convex, it is liable to local convergence, and thus there is no theoretical guarantee to find the global minimum. However, by initializing  $\mathbf{P}, \mathbf{t}$  by the solution obtained by one of the standard hashing methods, we have a good initial point that can be improved by network optimization,

## 5. Results

### 5.1. Data

In our experiments, we used the Rawseeds dataset [4, 10]. The dataset contained video sequences of a robot equipped with an omnidirectional camera system based on a parabolic mirror moving in an indoor and outdoor scene. The image undergoes significant distortion since different parts of the scene move from the central part of the mirror to the boundaries.

We used the toolbox of Vedaldi [40] to compute SIFT features in each frame of the video. Since the robot movement is slow, the change between two adjacent frames in the dataset is infinitesimal, and SIFT features can be matched reliably. Tracking features for multiple frames, we constructed the positive set as the transitive closure of these adjacent feature descriptor pairs. This way, the positive set included also descriptors distant in time, and, as a result of robot motion located at different regions in the image and thus subject to strong distortions. As negatives, we used features not belonging to the same track.

In addition to the Rawseeds dataset, we created synthetic omnidirectional datasets using panorama images that were warped simulating the effect of a parabolic mirror. The warping intentionally was not the same as in Rawseeds dataset. By moving the panorama image, we created synthetic motion with

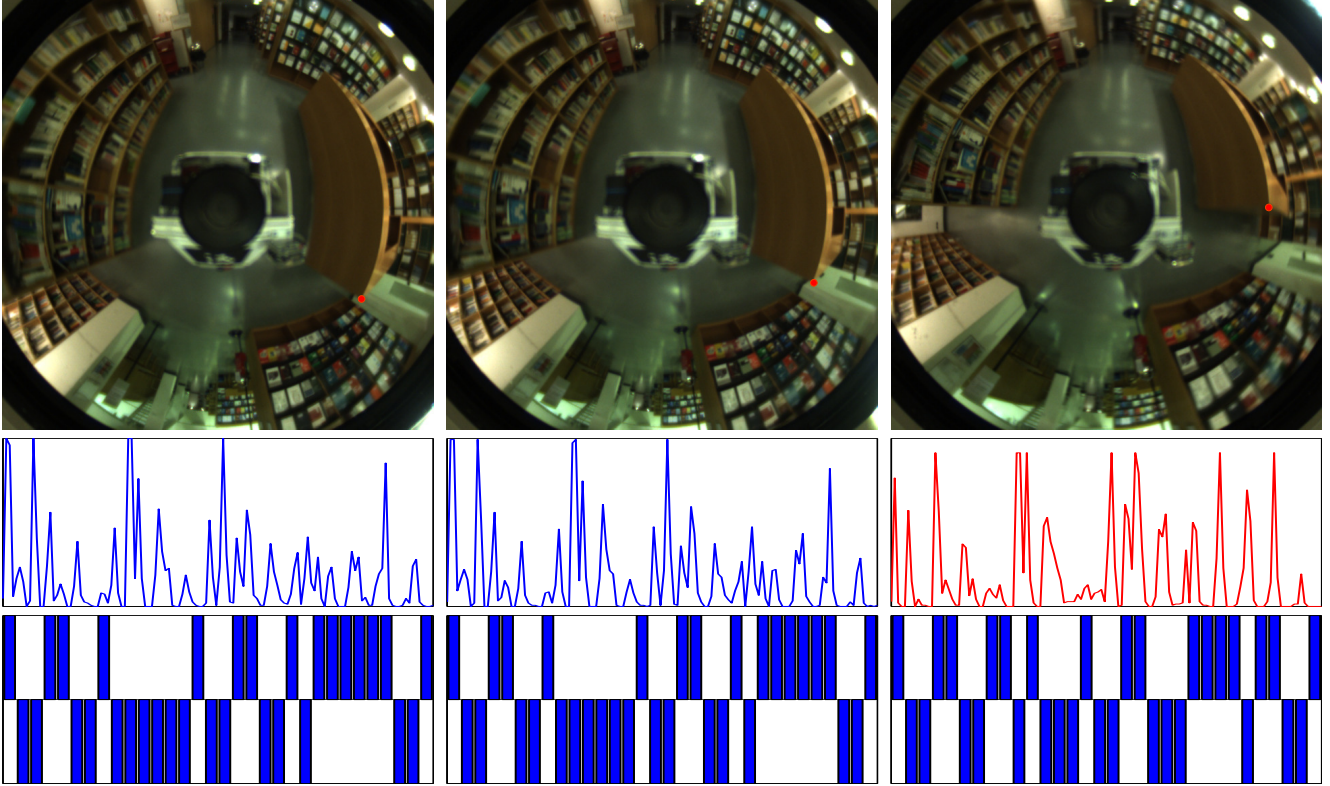


Figure 1: A few frames from the Rawseeds dataset exemplifying how a descriptor changes over time due to camera motion throughout the scene. First row: omnidirectional images of the indoor dataset, shown at times 1 (left), 5 (middle) and 50 (right). Second row: SIFT descriptors at point indicated in red. Third row: binary descriptors of length 32 produced by NNhash trained on outdoor images.

known pixel-wise groundtruth correspondence (Figure 5). The positive and negative sets for synthetic data were constructed as described above.

## 5.2. Methods

We compared the SSH [34], diff-hash [35], and our NNhash methods. For the NNhash training we used scaled conjugate gradient over the whole batch of descriptors, which we normalize in the range  $[-1..1]$ . We used a margin  $m = 5$  in all cases. The steepness factor for tanh is  $\beta = 1$  in the case of 32 bit while for 64 bit we gradually increased it up to 3 so to have a smooth binarization. We reached convergence in about 50 epochs in all cases.

## 5.3. Performance degradation in time

For this experiment, we constructed the training set using descriptors extracted from about 300 consecutive frames of the outdoor sequence (similar results were obtained when using outdoor or synthetic data for training). We considered descriptors that could be tracked for at least 60 consecutive frames and selected as positives pairs of descriptors belonging to these tracks.

To avoid bias, we selected pairs of descriptors in frames  $t_i, t_j$  in such a way that the time difference

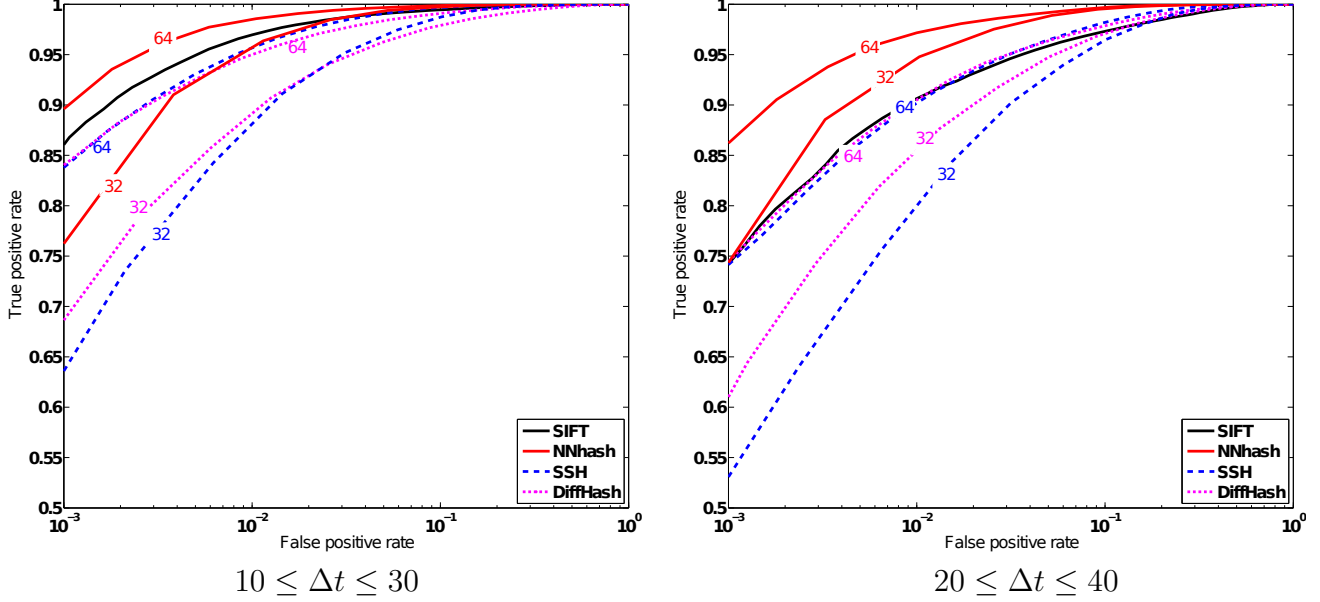


Figure 2: ROC curve for the outdoor dataset, with frames taken at various distance  $\Delta t$ . Each hashing method is shown with 32 and 64 bits. Note significant performance degradation of SIFT and only minor performance degradation of NNhash.

$\Delta t = |t_i - t_j|$  between the frames was uniformly distributed. The training was performed on a positive set of size  $10^5$  and on a negative set of size  $10^6$  to produce hashes of length 32 and 64 bits.

Testing was performed on a different portion of the same sequence, where frames at distance  $10 \leq \Delta t \leq 30$  (Figure 2, left) and  $20 \leq \Delta t \leq 40$  (Figure 2, right) were used. A few phenomena can be observed in Figure 2 showing the ROC curves of straightforward SIFT matching using the Euclidean distance and matching of learned binary descriptors using the Hamming distance. First, we can see that even with very compact descriptors (as small as 64 bit, compared to 1024 bit required to represent SIFT) we match or outperform SIFT. These results are consistent with the study in [35]. Second, we observe that NNhash significantly outperforms other hashing methods for the same number of bits. This is a clear indication that SSH and diff-hash methods are finding a suboptimal solution by solving a relaxed problem, while NNhash attempts to solve the full non-linear non-convex optimization problem.

Comparing Figure 2 (left and right) and Tables 1–2, we can observe how the matching performance degrades if we increase the time between the frames (from 10 – 30 frames to 20 – 40 frames). Because of significant distortions caused by the parabolic mirror, objects moving around the scene appear differently. This phenomenon is especially noticeable when the distance between the frames ( $\Delta t$ ) is large. SIFT shows significant degradation, while NNhash, trained on a dataset including positive pairs at distances up to  $\Delta t = 60$  degrades only slightly (even a 32-bit NNhash performs better than SIFT). This is a clear indication that we are able to learn feature invariance.

Finally, Figure 4 shows a visual example of feature matching using different methods. NNhash produces matches most similar to the groundtruth (shown in green).

	$m$	<b>EER</b>	<b>FPR@1%</b>	<b>FPR@0.1%</b>
<b>SIFT</b>	1024	1.91%	3.08%	13.87%
<b>NNhash</b>	32	<b>1.66%</b>	3.77%	23.81%
	64	<b>1.31%</b>	<b>1.92%</b>	<b>9.48%</b>
<b>DiffHash</b>	32	4.41%	9.36%	29.95%
	64	2.57%	5.17%	18.30%
<b>SSH</b>	32	4.02%	15.64%	36.41%
	64	2.22%	4.90%	16.74%

Table 1: Descriptor matching performance using different methods and descriptor size for frames with time range  $10 \leq \Delta t \leq 30$ .

	$m$	<b>EER</b>	<b>FPR@1%</b>	<b>FPR@0.1%</b>
<b>SIFT</b>	1024	3.31%	7.47%	27.94%
<b>NNhash</b>	32	<b>2.70%</b>	<b>6.98%</b>	<b>24.98%</b>
	64	<b>2.38%</b>	<b>4.54%</b>	<b>14.22%</b>
<b>DiffHash</b>	32	5.17%	12.55%	37.49%
	64	3.69%	8.75%	27.34%
<b>SSH</b>	32	5.52%	24.10%	47.29%
	64	3.46%	9.48%	27.66%

Table 2: Descriptor matching performance using different methods and descriptor size for frames with time range  $20 \leq \Delta t \leq 40$ .

#### 5.4. Generalization

To test for generalization we perform experiments of transfer learning from outdoor data to indoor data and from synthetic data to real data.

Figure 3-left shows the performance of descriptors trained on outdoor and tested on indoor data. We can see that even though the data used for training is very different from the one used for testing (i.e. see Figure 1 and Figure 4 for a visual comparison) we achieve better performance than SIFT with just 64 bits. Figure 3-right shows the performance of descriptors trained on synthetic and tested on indoor data. All learning methods perform better than SIFT. The discrepancy between NNhash and the other algorithms is less pronounced than in the real case.

## 6. Discussion, Conclusions, and Future Work

We presented a new approach for feature matching in omnidirectional images based on similarity-sensitive hashing and inspired by the recent work [35]. We learn a mapping from the descriptor space to the space of binary vectors that preserves the similarity of descriptors on a training set. By carefully constructing the training set, we account for descriptor variability, e.g. due to optical distortions. The resulting descriptors are compact and are compared using the Hamming metric, offering significant computational advantage over other traditional metrics such as  $L_2$ . Though tested with SIFT descriptors, our approach is generic and can be applied to any feature descriptor.

We compared several existing similarity-preserving hashing methods, as well as our NNhash method



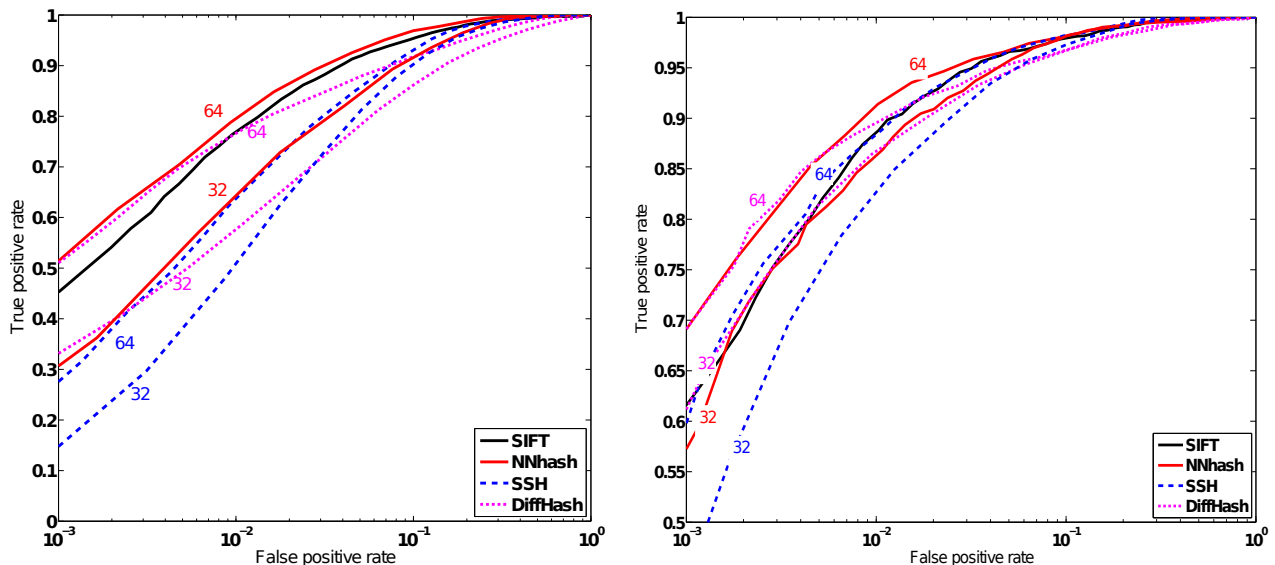


Figure 3: Left: ROC curve for the models trained on outdoor data and tested on indoor data with descriptors taken at  $35 \leq \Delta t \leq 60$ . Right: ROC curve for synthetic trained models. Testing performed on indoor real descriptors.

based on a neural network. Experimental results show that NNhash outperforms other approaches. An explanation to this behavior is the fact that of today’s state-of-the-art similarity-preserving hashing algorithms like SSH or LDAHash solve a simplified optimization problem, whose solution does not necessarily coincide with the solution of the “true” non-linear non-convex problem. We showed that using a neural network, we can solve the “true” problem and yield better performance.

Finally, our discussion in this paper was limited to simple embeddings of the form  $\text{sign}(\mathbf{P}\mathbf{x} + \mathbf{t})$  which in some cases are too simple. The neural network framework seems to us a very natural way to consider more generic embeddings using multi-layer network architectures.

## Acknowledgement

M. B. is partially supported by the Swiss High Performance and High Productivity Computing (HP2C) grant. J. M. is supported by Arcelor Mittal Maizières Research SA. D. M. is partially supported by the EU projects FP7-ICT-IP-231722 (IM-CLeVeR).

## References

- [1] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: a method for efficient approximate similarity ranking. In *Proc. CVPR*, 2004.
- [2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*, 10(3):346–359, 2008.
- [3] I. Bogdanova, X. Bresson, J. P. Thiran, and P. Vandergheynst. Scale space analysis and active contours for omnidirectional images. *Trans. Image Processing*, 16(7):1888–1901, 2007.

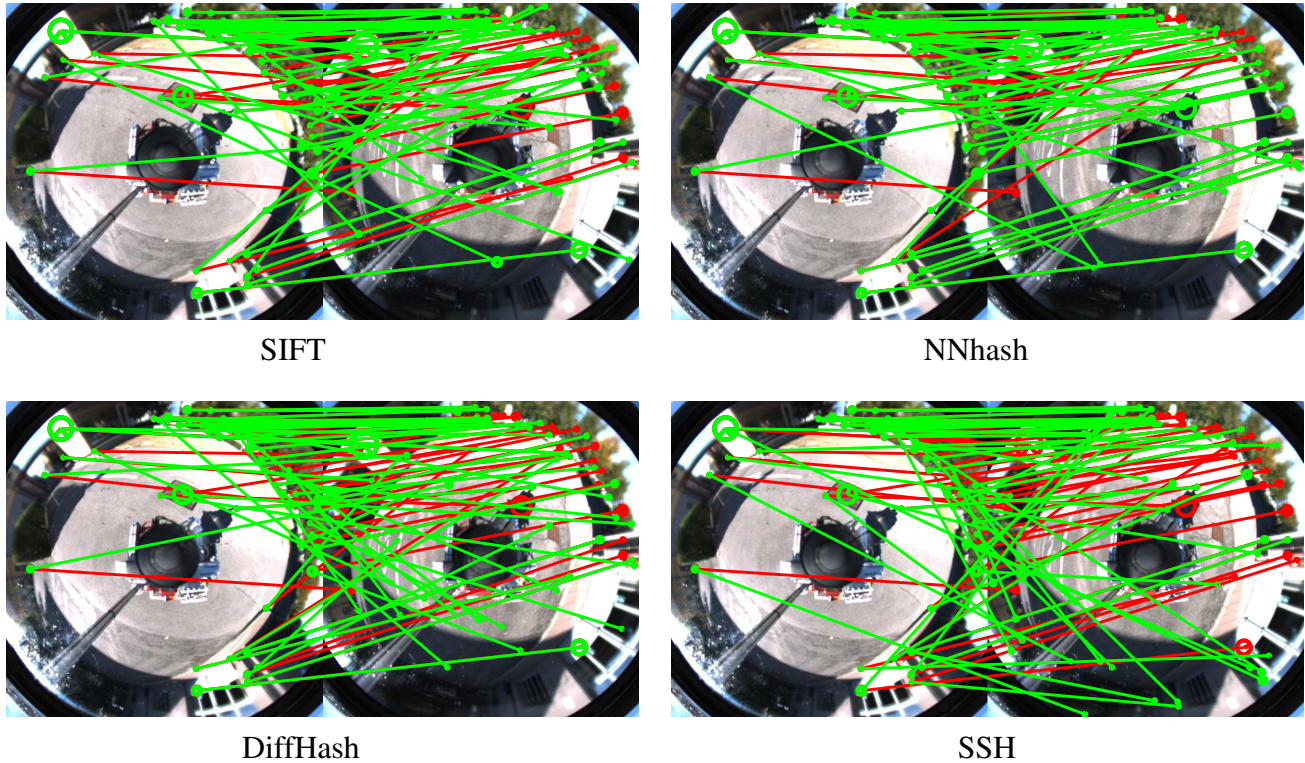


Figure 4: Visual comparison of the matches produced on outdoor data with  $\Delta t = 70$ . Ground truth matches are plotted in red and descriptor matches (1-closest) in green. Ideally (if matching completely coincides with the groundtruth), only green lines should be visible. Interesting matches appear on the bottom-left portion of the image where NNhash learns invariance to high distortions.

- [4] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos. Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets. In *Proc. IROS Workshop on Benchmarks in Robotics Research*, 2006.
- [5] A. Briggs, Y. Li, D. Scharstein, and M. Wilder. Robot navigation using 1d panoramic images. In *Proc. ICRA*, 2006.
- [6] A. Bronstein, M. Bronstein, M. Ovsjanikov, and L. Guibas. Shape Google: geometric words and expressions for invariant shape retrieval. *ACM TOG*, 2010.
- [7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Video genome. Technical Report arXiv:1003.5320v1, 2010.
- [8] M. M. Bronstein. Kernel diff-hash. Technical Report arXiv:1111.0466v1, 2011.
- [9] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *Proc. CVPR*, 2010.
- [10] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27(4):353–371, 2009.
- [11] J. Cruz, I. Bogdanova, B. Paquier, M. Bierlaire, and J. P. Thiran. Scale invariant feature transform on the sphere: Theory and applications. Technical report, 2009.

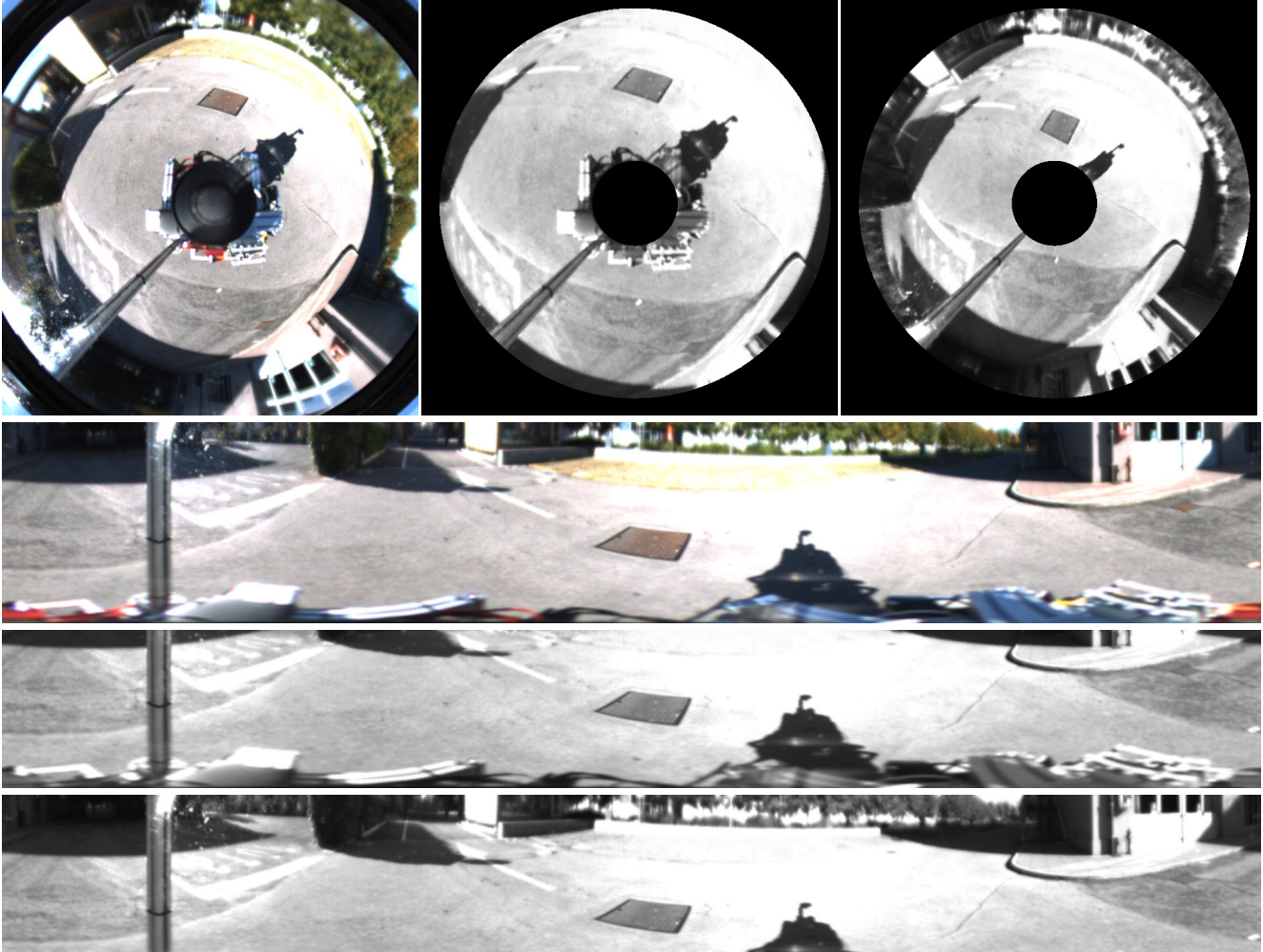


Figure 5: Illustrative example of how synthetic data is generated. First row from left to right: the original omnidirectional image, the synthetic image from the first shift of 5 pixels, the synthetic image after 14 vertical shifts. Second to fourth rows: unwrapped panorama images generated from images in the first row.

- [12] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.
- [13] C. Geyer and H. Stewenius. A nine-point algorithm for estimating para-catadioptric fundamental matrices. In *Proc. CVPR*, 2007.
- [14] A. Gionis, P. Indik, and R. Motwani. Similarity Search in High Dimensions via Hashing. In *International Conference on Very Large Databases*, 2004.
- [15] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, 2006.

- [16] P. I. Hansen, P. Corke, and W. Boles. Wide-angle visual feature matching for outdoor localization. *Int. J. Robotics Research*, 29(2/3):267–297, February 2010.
- [17] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, 2008.
- [18] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, 2008.
- [19] H. Jégou, M. Douze, and C. Schmid. Packing Bag-of-Features. In *Proc. ICCV*, 2009.
- [20] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *Trans. PAMI*, 2010.
- [21] R. Kimmel, C. Zhang, A. M. Bronstein, and M. M. Bronstein. Are msfer features really interesting? *Trans. PAMI*, 32(11):2316–2320, 2011.
- [22] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Proc. NIPS*, pages 1042–1050, 2009.
- [23] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 20(2):91–110, 2004.
- [24] T. Mauthner, F. Fraundorfer, and H. Bischof. Region matching for omnidirectional images using virtual camera planes. *Technology*, 2006.
- [25] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *Proc. ICRA*, 2007.
- [26] B. Micusik and T. Pajdla. Structure from motion with wide circular field of view cameras. *Trans. PAMI*, 28(7):1135–1149, 2006.
- [27] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005.
- [28] S. K. Nayar. Catadioptric Omnidirectional Camera. In *Proc. CVPR*, 1997.
- [29] L. Puig and J. J. Guerrero. Scale space for central catadioptric systems. towards a generic camera feature extractor. In *Proc. ICCV*, 2011.
- [30] M. Raginsky and S. Lazebnik. Locality-Sensitive Binary Codes from Shift-Invariant Kernels. In *Proc. NIPS*, 2009.
- [31] R. Salakhutdinov and G. Hinton. Semantic hashing. In *SIGIR Workshop on Information Retrieval and applications of Graphical Models*, 2007.
- [32] D. Scaramuzza, R. Siegwart, and A. Martinelli. A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics. *Int. J. Robotics Research*, 28(2):149–171, 2009.
- [33] J. Schmidhuber and D. Prelinger. Discovering predictable classifications. *Neural Computation*, 5(4):625–635, 1993.
- [34] G. Shakhnarovich. *Learning Task-Specific Similarity*. PhD thesis, MIT, 2005.
- [35] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua. LDAHash: improved matching with smaller descriptors. *Trans. PAMI*, 2011.
- [36] T. Svoboda and T. Pajdla. Matching in catadioptric images with appropriate windows, and outliers removal. In *Proc. CAIP*, 2001.

- [37] G. W. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invariance through imitation. In *Proc. CVPR*, 2011.
- [38] E. Tola, V. Lepetit, and P. Fua. Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo. *Trans. PAMI*, 32(5):815–830, 2010.
- [39] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *Trans. PAMI*, 30(11):1958–1970, 2008.
- [40] A. Vedaldi. An open implementation of the SIFT detector and descriptor. Technical Report 070012, UCLA CSD, 2007.
- [41] J. Wang, S. Kumar, and S. F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, 2010.
- [42] J. Wang, S. Kumar, and S. F. Chang. Sequential projection learning for hashing with compact codes. In *ICML*, 2010.
- [43] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. 2009.